Attorney Docket No. ANDIP001

PATENT APPLICATION

METHODS AND APPARATUS FOR ENCAPSULATING A FRAME FOR TRANSMISSION IN A STORAGE AREA NETWORK

Inventors:

Thomas James Edsall
13208 Peacock Court
Cupertino, CA 95014
Citizenship: U.S.

Dinesh Ganapathy Dutt
1176 Corral Ave
Sunnyvale, CA 94086
Citizenship: Indian

Silvano Gai
3021 Mauna Loa Ct
San Jose, CA 95132
Citizenship: Italian

Assignee:

Andiamo Systems Inc.
375 East Tasman Drive
San Jose, CA 95134

A Delaware corporation

Status: Large Entity

Prepared by:

BEYER, WEAVER & THOMAS, LLP

METHODS AND APPARATUS FOR ENCAPSULATING A FRAME FOR TRANSMISSION IN A STORAGE AREA NETWORK

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to network technology. More particularly, the present invention relates to methods and apparatus for encapsulating a frame for transmission in a storage area network.

2. Description of the Related Art

When interconnecting computers and other devices in a network, it has become desirable to create "virtual local area networks" (VLANs), in which all devices coupled to a VLAN receive all frames or packets which are universally addressed (whether by broadcast, multicast, or some other technique) on that VLAN, and in which all frames or packets which are universally addressed by a device on a VLAN are not distributed to devices outside the VLAN. The VLAN approach is particularly desirable when a single physical infrastructure is to be made available to multiple parties, each requiring that its data be kept private from the other parties. Further, the VLAN approach protects network entities on a given VLAN from failures of devices on the same infrastructure but belonging to a different VLAN.

Various VLAN transport protocol technologies have been proposed and come to

be accepted in the art. For example, VLAN technologies which are now common include LANE (for ATM LAN-Emulation), IEEE Standard 802.10, and various proprietary schemes such as Inter-Switch Links (ISL) (e.g., for Cisco Catalyst.TM. Inter-Switch Links).

In order to allow multiple VLANs to share a single inter-switch link on the underlying physical topology, the interswitch link protocol (ISL) was developed at Cisco Systems. See for example U.S. Pat. No. 5,742,604, entitled "Interswitch link mechanism for connecting high-performance network switches," Edsall, et al., issued on April 21, 1998 to Cisco Systems, Inc., which is hereby incorporated by reference for all purposes. ISL provides an encapsulation mechanism for transporting packets between ports of different switches in a network on the basis of VLAN associations among those ports

Although ISL supports multiple VLANs on a single underlying network topology, certain limitations have been observed. Some of these limitations prevent easy implementation of ISL on modern storage area networks (SANs).

In recent years, the capacity of storage devices has not increased as fast as the demand for storage. Therefore a given server or other host must access multiple, physically distinct storage nodes (typically disks). In order to solve these storage limitations, the storage area network was developed. Generally, a storage area network is a high-speed special-purpose network that interconnects different data storage devices and associated data hosts on behalf of a larger network of users. However, although a SAN enables a storage device to be configured for use by various network devices and/or entities within a network, data storage needs are often dynamic rather than static.

A SAN may use various types of network traffic such as Ethernet or Fibre Channel frames. Regardless of the technology used, current SAN technology requires

that a single protocol (e.g., Fibre Channel) be used throughout a particular SAN. Thus, current technology fails to address the need for supporting a multiple SAN system in which different transport protocols or technologies simultaneously co-exist. Note that ISL was originally designed for encapsulation of Ethernet packets. It does not support multiple different protocols on a single physical network infrastructure.

Ethernet is currently the most widely-installed LAN technology. The most commonly installed Ethernet systems are 10BASE-T systems, which provide transmission speeds up to 10 Mbps. Alternatively, fast Ethernet systems, 100BASE-T systems, provide transmission speeds up to 100 megabits per second, while Gigabit Ethernet provides support at 1000 megabits per second (or 1 billion bits per second).

While Ethernet is widely used, use of fibre channel is proliferating in systems which demand high bandwidth and low latency. More specifically, the fibre channel family of standards (developed by the American National Standards Institute (ANSI)) defines a high speed communications interface for the transfer of large amounts of data between a variety of hardware systems such as personal computers, workstations, mainframes, supercomputers, storage devices and servers that have fibre channel interfaces. Fibre channel is particularly suited for connecting computer servers to shared storage devices and for interconnecting storage controllers and drives. Moreover, fibre channel is capable of transmitting data between computer devices at a data rate of up to 1 Gbps (or 1 billion bits per second), and a data rate of 10 Gbps has been proposed by the Fibre Channel Industry Association. Accordingly, fibre channel is a technology that is in widespread use for transmitting data in SANs. However, as indicated above, ISL was not optimized for fibre channel transmissions.

In view of the above, it would be desirable if properties of a VLAN could be

merged with those of a SAN to enable various storage devices to be logically assigned to various entities within a network. Moreover, it would be beneficial if a single switching mechanism could simultaneously support different transport protocols (including at least fibre channel) within a network such as a SAN.

SUMMARY OF THE INVENTION

Methods and apparatus for encapsulating a packet or frame for transmission in a storage area network are disclosed. More particularly, according to the disclosed encapsulation process, a packet or frame is encapsulated with a virtual storage area network (VSAN) identifier. Through generation and transmission of such an encapsulated packet or frame, implementation of a single VSAN as well as a network including multiple interconnected VSANs may be achieved.

In accordance with various embodiments of the present invention, an encapsulation mechanism is implemented in a virtual storage area network (VSAN). Through the concept of a VSAN, one or more network devices (e.g., servers) and one or more data storage devices are grouped into a logical network defined within a common physical infrastructure. Each VSAN is uniquely identified by a VSAN identifier.

In accordance with one aspect of the invention, a packet or frame compatible with a standard protocol employed in the storage area network is received or generated. The packet or frame is then encapsulated with a VSAN identifier. For instance, a new header (or trailer) may be appended to the packet or frame compatible with the standard protocol. Once encapsulated, the encapsulated packet or frame is

sent over the storage area network. The encapsulated packet or frame is typically sent over a link such as an enhanced interswitch link shared by multiple VSANs.

In addition to the VSAN identifier, the encapsulated packet or frame may include further information. More particularly, in accordance with one embodiment, the packet or frame is also encapsulated with at least one of a Time To Live (TTL) value and/or Multi-Protocol Label Switching (MPLS) information. For instance, the TTL value may be used to specify a number of remaining hops that can be traversed before the encapsulated packet or frame is dropped. The TTL value may also be used to specify a remaining lifetime in units of time (e.g. milliseconds). MPLS is a common forwarding mechanism used in various technologies to forward packets and frames such as IP packets and Ethernet frames. However, MPLS has not been implemented or proposed for use with Fibre Channel frames.

In accordance with another embodiment, the frame is encapsulated with a type of traffic to be carried by the frame. For instance, the type of traffic to be carried by the frame may include Ethernet, Fibre Channel, and Infiniband. Typically, this "type" refers to the standard protocol employed to generate the frame in question. Through the identification of a traffic type, frames carrying a variety of traffic types may be transmitted within a VSAN. Moreover, multiple VSANs, each capable of supporting different traffic types, may be interconnected through the identification of a traffic type in the newly appended header.

Various network devices may be configured or adapted for generation of a frame compatible with a standard protocol (e.g., Fibre Channel). Similarly, a variety of network devices may be capable of receiving the frame, encapsulating the frame,

and sending the encapsulated frame via a VSAN. These network devices include, but are not limited to, servers (e.g., hosts) and switches. Moreover, the functionality for the above-mentioned generation and encapsulation processes may be implemented in software as well as hardware.

Yet another aspect of the invention pertains to computer program products including machine-readable media on which are provided program instructions for implementing the methods and techniques described above, in whole or in part. Any of the methods of this invention may be represented, in whole or in part, as program instructions that can be provided on such machine-readable media. In addition, the invention pertains to various combinations and arrangements of data generated and/or used as described herein. For example, encapsulated frames having the format described herein and provided on appropriate media are part of this invention.

These and other features of the present invention will be described in more detail below in the detailed description of the invention and in conjunction with the following figures.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a diagram illustrating an exemplary storage area network, including multiple VSANs, in which the present invention may be implemented.

FIG. 2 is a diagram illustrating an "extension" to a frame such as a fibre channel in accordance with one embodiment of the invention.

FIG. 3 is a diagram illustrating an extended ISL (EISL) frame format including an EISL header in accordance with one embodiment of the invention.

FIG. 4 is a diagram illustrating an exemplary EISL header (and associated MPLS label stack) that may be transmitted in a frame having an EISL frame format such as that illustrated in FIG. 3.

FIG. 5 is a diagram illustrating an exemplary MPLS label format that may be used for each label in the MPLS label stack of FIG. 4 in accordance with one embodiment of the invention.

FIG. 6 is a diagram illustrating an exemplary network device in which embodiments of the invention may be implemented.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be obvious, however, to one skilled in the art, that the present invention may be practiced without some or all of these specific details. In other instances, well known process steps have not been described in detail in order not to unnecessarily obscure the present invention.

As described above, a storage area network (SAN) is a network that interconnects different data storage devices with associated network hosts (e.g., data servers or end user machines) on behalf of a larger network of users. A SAN is defined by the physical configuration of the system. In other words, those devices in a SAN must be physically interconnected.

In accordance with various embodiments of the present invention, the physical SAN is shared by multiple virtual storage area networks (VSANs), each using some or all of the physical infrastructure of the SAN. An encapsulation mechanism is employed to implement the VSANs. Through the concept of a VSAN, one or more network devices (e.g., servers) and one or more data storage devices are grouped into a logical network defined within a common physical network infrastructure. Communication among these devices may be accomplished by coupling the network devices and data storage devices together through one or more switches. Exemplary data storage devices that may be used in a VSAN include, but are not limited to, storage disks in various configurations such as a redundant array interconnected disk (RAID).

Each VSAN is uniquely identified by a VSAN identifier rather than any particular physical configuration of devices within the network. For instance, a unique identifier may be simultaneously associated with one or more network devices as well as one or more storage devices. In general, the VSAN identifier does not represent any particular physical network device or link.

Within a VLAN infrastructure, individual VLANs are distinguished through the use of a VLAN identifier (e.g., within the range of zero through 4094). In accordance with various embodiments of the invention, individual VSANs will be distinguished within a VSAN infrastructure through the use of a VSAN identifier, as described above. It would be desirable to support mixed infrastructures containing both VLANs and VSANs, for example, using a trunk link or router able to carry traffic among VLANs and VSANs.

In accordance with one embodiment, VLAN identifiers and VSAN identifiers share the same number space. In other words, any given number can be a VSAN identifier or a VLAN identifier, but cannot be used to independently identify a VSAN and a VLAN at the same time. For example, it would not be possible to have both a VSAN #1 and VLAN #1 at the same time.

In accordance with another embodiment, VLAN identifiers and VSAN identifiers have independent number spaces. In other words, a given number can be either a VSAN identifier or a VLAN identifier, as well as an identifier for both a VSAN and a VLAN simultaneously. Since a number may correspond to either a VSAN or a VLAN, the type of network (e.g., VSAN or VLAN) is also identified. Thus, VSAN #1 and VLAN #1 can both exist at the same time as different and

independent virtual networks. This type may be a separate field in the EISL header or may be incorporated into the frame type (e.g., traffic type), which will be described in further detail below with reference to FIG. 4.

FIG. 1 is a diagram illustrating an exemplary storage area network 101 in which the present invention may be implemented. As shown in FIG. 1, data storage devices 102, 104, 106, 108, 110, and 112 are coupled to hosts (e.g., servers) 114, 116, and 128 via several switches 118, 120, and 122. The switches communicate with one another via interswitch links 124 and 126. The elements described in this paragraph together comprise the physical infrastructure of SAN 101. The storage devices, switches, and hosts of SAN 101 communicate using one or more standard protocols such as fibre channel (FC), fibre channel IP (FCIP), SCSI, SCSI over IP, Ethernet, Infiniband, and the like. Sometimes, these protocols are referred to herein as "types" of traffic.

One or more VSANs may be created through logically grouping various network devices with selected data storage devices. For instance, as shown in FIG. 1, a first VSAN, VSAN1, consists of server 114, 128 and data storage devices 102, 106, 110, and 114, while a second VSAN, VSAN2, consists of server 116, 128 and data storage devices 104, 108, and 112. Note that both VSAN1 and VSAN2 share the interswitch links 124 and 126 to communicate. For example, both host 114 of VSAN1 and host 116 of VSAN2 use link 126 to access storage device 110 and storage device 112, respectively.

To ensure that network traffic crossing interswitch links 124 and 126 is properly routed to (and limited to) devices within the associated VSANs, the traffic

must be identifiable as either VSAN1 traffic or VSAN2 traffic. This can be accomplished in various ways. One convenient methodology employed with this invention encapsulates data frames with some sort of VSAN identifier. That way switches 118, 120, and 122 can examine frames crossing links 124 and 126 to determine which VSAN is involved and make appropriate switching decisions based upon this information.

Various embodiments of the invention may be applied to encapsulate a packet or frame as described below with reference to FIGs. 2 through 5. The distinction between packet and frame is not universally accepted. However, one distinction that is commonly used is defined as follows. A packet is the unit of data that is routed between an origin and a destination on the Internet or any other packet-switched network. Each packet is separately numbered and includes the Internet address of the destination. In contrast, a frame is data that is transmitted between. adjacent network devices. The information or data in the frame may contain packets or other data units used in a higher-level or different protocol.

For convenience, the subsequent discussion will describe encapsulated *frames* used to transmit data over SAN. Switches act on frames and use information about VSANs to make switching decisions. Thus, in accordance with various embodiments of the invention, frames that originate within a VSAN are constrained to stay within the physical resources allotted to that VSAN. Thus, a gateway working at a higher protocol layer (e.g., application layer 7) may pass such a packet from one VSAN to another. In order to enforce this restriction, frames may be encapsulated with a structure that identifies a VSAN to which the frames belong. Once encapsulated, the

encapsulated frame is moved across network nodes; in particular over inter-switch links in the storage area network. Thus, through this encapsulation, a single link may transport frames associated with multiple VSANs. An exemplary encapsulated packet will be described in further detail below with reference to FIG. 3 and FIG. 4.

Note that the frames being encapsulated possess the frame format specified for a standard protocol such as Ethernet or fibre channel. Hence, software and hardware conventionally used to generate such frames may be employed with this invention. Additional hardware and/or software is employed to encapsulate the standard frames in accordance with this invention. Those of skill in the art will understand how to develop the necessary hardware and software to allow encapsulation of the type described below.

In accordance with the present invention, a frame compatible with a standard protocol employed in a storage area network is obtained at a node where encapsulation subsequently takes place. The frame is generated by a network device such as a host, switch, or storage device. In a typical scenario, the host machine generates the frame in accordance with a standard protocol such as fibre channel and forwards the frame to a switch. The switch then performs an encapsulation process such as that described below. In some embodiments, the host or storage device may itself perform the encapsulation described herein.

In still other embodiments, the switch may receive a data stream and first frame it in accordance with a standard protocol and then encapsulate the resulting frame to produce the new frames of this invention. The resulting encapsulated frame is sometimes referred to herein an "extended ISL" (EISL) frame because of its

general relationship with the interswitch link protocol described in US Patent No. 5,742,604, previously incorporated by reference.

As just explained, the encapsulation process may be performed by a network device outside the switch (e.g., by a host) as well as within a switch. Obviously, the appropriate network devices should be configured with the appropriate software and/or hardware for performing EISL encapsulation. Of course, all network devices within the storage area network need not be configured with EISL encapsulation software (or hardware). Rather, selected network devices may be configured with or adapted for EISL encapsulation functionality. Similarly, in various embodiments, such EISL encapsulation functionality may be enabled or disabled through the selection of various modes. Moreover, it may be desirable to configure selected ports of network devices as EISL-capable ports capable of performing EISL encapsulation, either continuously, or only when in an EISL enabled state.

The standard protocol (e.g., layer 2 protocol) employed in the storage area network (i.e., the protocol used to frame the data) will typically, although not necessarily, be synonymous with the "type of traffic" carried by the network. As explained below, the type of traffic is defined in some encapsulation formats. Examples of the type of traffic are typically layer 2 or corresponding layer formats such as Ethernet and Fibre Channel.

Another type of traffic that may be used within a VSAN is an InfiniBand architecture, which is a relatively recent input/output (I/O) specification for servers. More particularly, connections between servers, remote storage and networking are accomplished by attaching all devices through a central, unified, fabric of InfiniBand

switches and links. Stated benefits of the InfiniBand architecture include lower latency, easier and faster sharing of data, and built in security and quality of service. The InfiniBand architecture is a rapidly developing technology for server clusters as well as I/O for remote storage and networking. Thus, the InfiniBand Architecture is well suited to address the demands generated by the rapid growth of the Internet and the convergence of data and telecommunications (voice, data, video, and storage) on the Internet. With InfiniBand, data is transmitted in one or more packets that together form a message. For instance, a message can be a remote direct memory access (RDMA) read or write operation, a channel send or receive message, a transaction-based operation (that can be reversed), or a multicast transmission.

In addition, other types of traffic may include token ring, token bus, and various protocols compatible with satellite systems such as Aloha. It is also important to note that the type of traffic should also be supported by the technology used in the storage area network. For instance, the technology used in the hosts and/or switches (e.g., interface port circuitry) as well as the communication mediums should support the traffic type being carried in the encapsulated frame.

As shown in FIG. 2, a frame compatible with a standard protocol employed in a storage area network is obtained. For instance, a frame such as a fibre channel frame 202 typically includes a header 204, payload 206 and error check information such as a cyclic redundancy checking (CRC) value 208. Cyclic redundancy checking is a method of checking a received frame for errors in data that has been transmitted on a communications link. As described above, it is necessary to identify the VSAN associated with the frame 202. However, pursuant to the standard governing fibre channel, the header 204 of a fibre channel frame 202 does not include unused bits to

enable additional fields to be defined. As a result, an "extension" 210 to the frame 202 is required. Extension 210 may assume various formats. Importantly, it should include at least information identifying the VSAN and other information specifying one or more of the following: type of traffic, MPLS information, and time to live.

In accordance with various embodiments, as shown in FIG. 3, a new frame 302 having an extended ISL (EISL) format is generated from information obtained from the original frame 202. More particularly, encapsulation may include the appending of a new header (or trailer) to the original frame 202. For instance, as shown, the new frame 302 includes a payload 303 that preferably includes both the header 204 and payload 206 of the original frame, and a new EISL header 304. In addition, the payload 303 may also include the CRC 208 of the original frame. Encapsulation may also comprise the modification or replacement of the original CRC value 208 with a new (e.g., additional) or modified CRC value 306. More particularly, since the new frame is longer than the original frame 202, a new CRC value 306 may be generated (e.g., calculated) to correspond the longer length of the frame 302 that includes the newly appended EISL header 304 and its associated length. This new CRC 306 may therefore replace the original CRC 208. In this manner, a new frame is generated by encapsulating the frame with an EISL header.

In accordance with various embodiments, an EISL frame 302 such as that illustrated in FIG. 3 is preceded by a start of frame (SOF) delimiter 308 and terminated by an end of frame (EOF) delimiter 310. These delimiters enable an EISL-capable port to receive and recognize frames in EISL format. However, if an EISL-capable port is not in EISL mode or, alternatively, the port is not EISL-capable and it

receives frames in the EISL format, it can drop the frame.

FIG. 4 is a diagram illustrating an exemplary EISL header of a frame that has an overall EISL frame format such as that illustrated in FIG. 3. More particularly, as described above, the EISL header preferably identifies a VSAN. In addition, encapsulation may include providing further information in the encapsulated frame. As will be described in further detail below, the EISL header may supply further information in various fields of the EISL header. More particularly, an EISL indicator field 402 may be used to indicate the presence of an EISL header. In addition, an EISL version indicator field 404 may indicate a version of EISL used to create the frame. In a specific example, the EISL version indicator field 404 of the EISL header includes at least 2 bits.

The EISL header further includes a field indicating a frame type 406 (i.e., traffic type). The type of traffic (e.g., payload) to be carried by the frame may include a variety of traffic types such as those described above, including but not limited to, Ethernet, fibre channel, and Infiniband. In one embodiment, the frame type field 406 is a 4-bit field. Through the identification of a traffic type, EISL formatted frames carrying a variety of traffic types may be transmitted within a VSAN. Moreover, multiple VSANs, each capable of supporting different traffic types, may be interconnected through the identification of a traffic type in the newly appended EISL header.

Multi-Protocol Label Switching (MPLS) label field (e.g., Indicator) 408 indicates whether the EISL header is carrying MPLS information such as an MPLS label stack, a common forwarding mechanism for both fibre channel and Ethernet

frames. An exemplary MPLS label stack will be described in further detail below with reference to FIG. 5. In one embodiment, the MPLS label field 408 is a 1-bit field. More particularly, the MPLS label field 408 indicates whether or not MPLS labels are provided in the EISL frame. For instance, the indicator may be set to 1 if the EISL header includes an MPLS label stack and otherwise be set to 0. The MPLS label field 408 may also indicate the number of labels present in 416.

Priority field 410 may indicate a user priority for the EISL frame. The user priority may represent various types of priorities. As one example, the user priority may be a generic priority, without a guaranteed level of service, that is used to merely indicate a priority such as a numerical ranking. For instance, higher values simply represent higher user priority while lower values may represent lower priority. Higher priority users receive available bandwidth first, regardless of how much total bandwidth is available. The number of bits used for this field will vary with the number of priority levels or values implemented.

As another example, the user priority may indicate a quality of service (QoS) of the payload of the EISL frame. On the Internet and in other networks, QoS is the idea that transmission rates, error rates, and other characteristics can be measured, improved, and, to some extent, guaranteed in advance. QoS is of particular concern for the continuous transmission of high-bandwidth video and multimedia information. Transmitting this kind of content dependably is difficult in public networks using ordinary "best effort" protocols. Typically, the number of bits required for a priority field are greater to specify the quality of service than a simple numerical priority. In one embodiment, the priority field 410 is a 3-bit field.

As described above, the EISL header, at a minimum, includes a VSAN

identifier field 412 adapted for including a VSAN identifier that identifies one or more VSANs. More particularly, in accordance with one embodiment, the VSAN identifier identifies a VSAN associated with the payload of the EISL frame, and therefore the payload of the original frame (e.g., Fibre Channel frame). In accordance with one embodiment, the VSAN identifier field 412 is a 12-bit field. The format of the identifier may be identical to or similar to VLAN identifiers as well as similar to addresses employed in certain standard protocols such as Ethernet.

In some storage area networks, there may be topology as well as routing problems that could cause a frame to traverse a loop within the network. Such a loop will consume bandwidth unnecessarily. In order to solve this problem, a Time To Live (TTL) field 414 may be used to indicate a TTL value specifying the number of remaining hops that can be traversed before the frame is dropped. More specifically, the TTL value of the TTL field 414 is initialized by the network device (e.g., switch) that generates the EISL frame including the EISL header. A default value may, for example, be set to 16. Subsequent network devices (e.g., switches) receiving the EISL frame decrement the TTL value of the TTL field 414 by 1.

In accordance with one embodiment, a TTL value of 1 indicates to the receiving network device (e.g., switch) that the EISL frame should be dropped. When the EISL frame is dropped, an error message may be sent to the intended recipient of the frame as well as to the sender of the frame. Similarly, a TTL value of 0 may indicate that the TTL field 414 should be ignored, allowing the EISL frame to be forwarded by the switch. In accordance with one embodiment, the TTL field 414 is an 8-bit field.

As described above, an MPLS indicator 408 may be used to indicate whether the EISL header is carrying MPLS information such as an MPLS label stack 416. As a packet of a connectionless network layer protocol travels from one router to the next, each router makes an independent forwarding decision for that packet. That is, each router analyzes the packet's header, and each router runs a network layer routing algorithm. Each router independently chooses a next hop for the packet, based on its analysis of the packet's header and the results of running the routing algorithm.

Packet headers contain considerably more information than is needed simply to choose the next hop. Choosing the next hop can therefore be thought of as the composition of two functions. The first function partitions the entire set of possible packets into a set of "Forwarding Equivalence Classes (FECs)." The second function maps each FEC to a next hop. Insofar as the forwarding decision is concerned, different packets which get mapped into the same FEC are indistinguishable. All packets which belong to a particular FEC and which travel from a particular node will follow the same path (or if certain kinds of multi-path routing are in use, they will all follow one of a set of paths associated with the FEC).

In conventional IP forwarding, a particular router will typically consider two packets to be in the same FEC if there is some address prefix X in that router's routing tables such that X is the "longest match" for each packet's destination address. As the packet traverses the network, each hop in turn reexamines the packet and assigns it to a FEC. In MPLS, the assignment of a particular packet to a particular FEC is done just once, as the packet enters the network. The FEC to which the packet is assigned is encoded as a short fixed length value known as a "label."

When a packet is forwarded to its next hop, the label is sent along with it; that

is, the packets are "labeled" before they are forwarded. At subsequent hops, there is no further analysis of the packet's network layer header. Rather, the label is used as an index into a table which specifies the next hop and a new label. The old label is replaced with the new label, and the packet is forwarded to its next hop. In the MPLS forwarding paradigm, once a packet is assigned to a FEC, no further header analysis is done by subsequent routers; all forwarding is driven by the labels. This has a number of advantages over conventional network layer forwarding. For instance, through MPLS labels, it is possible to circumvent conventional IP routing. Further details of the MPLS forwarding mechanism and MPLS label stack encoding are disclosed in RFC 3031, Multiprotocol Label Switching Architecture, Rosen et al., January 2001 and RFC 3032, MPLS Label Stack Encoding, Rosen et al., January 2001, respectively, and are incorporated by reference for all purposes.

As described above with reference to block 408 of FIG. 4, an indicator may be used to indicate whether the EISL frame includes a label stack. In accordance with one embodiment, a label stack may include a maximum of 4 labels. Each label in the label stack has an identical label format.

FIG. 5 is a diagram illustrating an exemplary MPLS label format that may be used for each label in the MPLS label stack of FIG. 4 in accordance with one embodiment of the invention. Label field 502 carries the actual value of the corresponding label used to make forwarding decisions. When a labeled packet is received, the label value at the top of the stack is looked up. As a result of a successful lookup in a corresponding table, file or database, the following information is ascertained: a) the next hop to which the packet is to be forwarded, and b) the operation to be performed on the label stack before forwarding; this operation may be

to replace the top label stack entry with another, to pop an entry off the label stack, or to replace the top label stack entry and then to push one or more additional entries on the label stack. In addition to learning the next hop and the label stack operation, one may also learn the outgoing data link encapsulation, and possibly other information which is needed in order to properly forward the packet. In accordance with one embodiment, the label field 502 is a 32-bit field.

Experimental field 504 may be reserved for experimental use. The experimental field 504 is typically used to encode the Differentiated Service Code Point (DSCP), which is a mechanism for Quality of Service.

In addition, a termination field 506 may be provided in the label to indicate if this label is the last label in the stack. In accordance with one embodiment, the termination field is a single bit indicator. For instance, the termination field 506 may indicate that the label is the last label in the label stack when the indicator is in a first state (e.g., 1) and that the label is not the last label in the label stack when the indicator is in a second state (e.g., 0).

The label may also include a TTL field 508. The TTL field 508 is typically used to provide TTL field semantics associated with IPv4 packets or Hop Count semantics associated with IPv6 packets.

As described above, encapsulation may be performed in a variety of network devices. FIG. 6 is a diagram illustrating an exemplary network device in which embodiments of the invention may be implemented. The network device illustrated is a hybrid switch that can switch both Ethernet and fibre channel frames. It is preferred that frames of various types (e.g., Ethernet and fibre channel) be transported via a

single switching mechanism. Through the use of an extended ISL (EISL) frame format, frames of different types may be transported using the same switch or ISL, rather than dedicating a switch or ISL to different frame (or traffic) types.

As shown in FIG. 6, data is received by a port 602 of the switch via a bi-directional connector (not shown). In association with the incoming port, Media Access Control (MAC) block 604 is provided, which enables frames of various protocols such as Ethernet or fibre channel to be received. Once a frame is encapsulated as described above, it is then received by a forwarding engine 608, which obtains information from various fields of the frame, such as source address and destination address. The forwarding engine 608 then accesses a forwarding table (not shown) to determine whether the source address has access to the specified destination address. The forwarding engine 608 also determines the appropriate port of the switch via which to send the frame, and generates an appropriate routing tag for the frame.

Once the frame is appropriately formatted for transmission, the frame will be received by a buffer queuing block 606 prior to transmission. Rather than transmitting frames as they are received, it may be desirable to temporarily store the frame in a buffer or queue 606. For instance, it may be desirable to temporarily store a packet based upon Quality of Service in one of a set of queues that each correspond to different priority levels. The frame is then transmitted via switch fabric 610 to the appropriate port. Each outgoing port also has its own MAC block and bi-directional connector via which the frame may be transmitted.

Although the network device described above with reference to FIG. 6 is

described as a switch, this network device is merely illustrative. Thus, other network devices such as routers may be implemented to receive, process, modify and/or generate packets or frames with functionality such as that described above for transmission in a storage area network. Moreover, the above-described network devices are merely illustrative, and therefore other types of network devices may be implemented to perform the disclosed encapsulation functionality.

Once the EISL frame is generated, the encapsulated frame may be sent by the switch over an inter-switch (ISL) link configured to couple source and destination port interface circuitry in a sequential arrangement of interconnected switches as illustrated in FIG. 1. The ISL link may consist of any type of media (e.g., twisted-pair or fiber) capable of functioning as an extension to a switching bus or other communication medium. Moreover, port interface circuitry may comprise encapsulation as well as decapsulation circuits. Since the interface to the ISL essentially comprises the port interface circuitry, any number of ports of a network device such as the switch illustrated in FIG. 6 are capable of being configured as EISL ports. That is, the ISL port interface circuit may include a circuit and/or otherwise be configured with software that includes functionality for encapsulating packets according to the disclosed EISL encapuslation mechanism, as well as decapsulating packets.

Although illustrative embodiments and applications of this invention are shown and described herein, many variations and modifications are possible which remain within the concept, scope, and spirit of the invention, and these variations would become clear to those of ordinary skill in the art after perusal of this

application. For instance, the present invention is described as being implemented using a specific EISL header. However, it should be understood that the invention is not limited to such implementations, but instead would equally apply regardless of the fields defined in such a header. Moreover, the present invention would apply regardless of the context and system in which it is implemented. Thus, broadly speaking, the operations described above need not be used within a SAN, but may be used to enable protocol compatibility within any network.

In addition, although an exemplary switch is described, the above-described embodiments may be implemented in a variety of network devices (e.g., servers) as well as in a variety of mediums. For instance, instructions and data for implementing the above-described invention may be stored on a disk drive, a hard drive, a floppy disk, a server computer, or a remotely networked computer. Accordingly, the present embodiments are to be considered as illustrative and not restrictive, and the invention is not to be limited to the details given herein, but may be modified within the scope and equivalents of the appended claims.